BACKWARD COMPATIBILITY THROUGH USE OF SPOOF CLOCK AND FINE GRAIN FREQUENCY CONTROL

CLAIM OF PRIORITY

[0001] This application is a continuation of U.S. patent application Ser. No. 16/740,271, filed Jan. 10, 2020 to be issued as U.S. Pat. No. 11,119,528. U.S. patent application Ser. No. 16/740,271 is a continuation of U.S. patent application Ser. No. 15/701,736 filed Sep. 12, 2017, now U.S. Pat. No. 10,534,395, the entire contents of which are incorporated herein by reference. U.S. patent application Ser. No. 15/701,736 is a continuation of U.S. patent application Ser. No. 14/627,988 filed Feb. 20, 2015, now U.S. Pat. No. 9,760,113, the entire contents of which are incorporated herein by reference.

FIELD OF THE DISCLOSURE

[0002] Aspects of the present disclosure are related to execution of a computer application on a computer system. In particular, aspects of the present disclosure are related to a system or a method that provides backward compatibility for applications/titles designed for older versions of a computer system.

BACKGROUND

[0003] Modern computer systems often use different processors for different computing tasks. In addition to a central processing unit (CPU), a modern computer may have a graphics processing unit (GPU) dedicated to certain computational tasks in a graphics pipeline, both being potentially part of an accelerated processing unit (APU) that may contain other units as well.

[0004] More powerful central processing units (CPUs), graphic processing units (GPUs) and accelerated processing units (APUs) may have higher latency, or latency characteristics that differ from less powerful components. For example, a more powerful GPU may have more stages in its texture pipeline when compared to a less powerful GPU. In such a case, the latency of this pipeline increases. In another example, a more powerful APU may contain a L3 cache for the CPU, compared to a less powerful APU that did not have such a cache. In such a case, the memory latency characteristics differ as the time needed to access data that misses all caches increases for the more powerful APU, but average latency will decrease for the more powerful APU.

[0005] The more powerful device and the less powerful device may be able to perform the same processing (e.g., execution of program instructions on the CPU or various programmatic and fixed function operations on the GPU), but differences in latency of this processing may cause the more powerful device to fail to be backwards compatible with respect to the less powerful device. Similarly, there may be differences in speed or throughput of the processing that cause the more powerful device to fail to be backwards compatible. For example, for certain types of processing, the more powerful device may be able to perform more iterations of the processing within the same time interval. Alternatively, the more powerful device could perform the processing using different algorithms that result in behavior that is faster or slower than the less powerful device, depending on the circumstance.

[0006] In the case of video game consoles, the operation is typically at a set clock frequency, and the software applications are tested for proper operation at this set frequency. Sometimes, it is desirable to run applications created for the original, less powerful console on a more powerful console. This ability is often referred to as "backward compatibility". In such cases, it is desirable for the more powerful device to be able to run the application created for the less powerful device without detrimental effects of differences in latency or processing speed.

[0007] It is within this context that aspects of the present disclosure arise.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] The teachings of the present disclosure can be readily understood by considering the following detailed description in conjunction with the accompanying drawings, in which:

[0009] FIG. **1** is a block diagram illustrating a system that may be configured at various operating frequencies in accordance with aspects of the present disclosure.

[0010] FIG. **2** is a flow diagram illustrating an example of a possible process flow in determining an operating frequency for a system in accordance with aspects of the present disclosure.

DESCRIPTION OF THE DRAWINGS

[0011] Although the following detailed description contains many specific details for the purposes of illustration, anyone of ordinary skill in the art will appreciate that many variations and alterations to the following details are within the scope of the invention. Accordingly, the exemplary embodiments of the invention described below are set forth without any loss of generality to, and without imposing limitations upon, the claimed invention.

Introduction

[0012] Several methods may be used for running applications created for the less powerful console on the more powerful console. In one example, the more powerful console may be set to run at the frequency of the original console. At this frequency setting, the operation of the more powerful console will vary based on the specific processing being performed at any instant of time, and may be slower or faster than the less powerful console due to the latency (and other) characteristics of that specific processing being performed. When the operation of the more powerful console is slower than the original console, many errors in the application may arise due to the inability to meet real time deadlines imposed by display timing, audio streamout or the like.

[0013] In another example, the more powerful console may be set to run at a much higher frequency than the original console. Speed of operation will vary based on the specifics of the processing being performed, but it will be consistently higher than on the original console and thus real time deadlines can be met successfully. However, many errors in the application may arise due to the untested consequences of such high speed operation. For example, in a producer-consumer model, if the consumer of data operates at higher speed than originally anticipated, it may attempt to access data before the data producer makes it available, and although synchronization mechanisms may